

## INFORME DE LABORATORIO

Autores: Angee Lorena Ocampo Ramírez, Oscar Andrés Gutiérrez Rivadeneira

Laboratorio de Electrónica Digital 2 Departamento de Ingeniería Electrónica y de Telecomunicaciones Universidad de Antioquia

## Resumen

En la presente práctica se busca implementar una Unidad Aritmética Lógica que contiene las operaciones de suma, resta, And y Or; esto se lleva a cabo por medio de un análisis basado en los conceptos adquiridos durante las sesiones de teoría y práctica, para, posteriormente, desarrollar el diseño mediante el lenguaje de descripción de hardware SystemVerilog, que brinda herramientas como el simulador ModelSim, para la evaluación del comportamiento del diseño antes de su implementación. Palabras clave: ALU, Compuertas lógicas, FPGA, Sumador, System Verilog

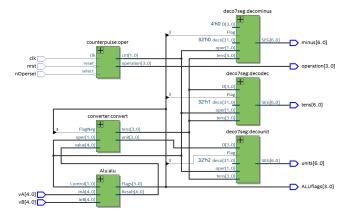


Figura **0-1**: Diagrama de bloques

# Diseño y procedimiento

A continuación se procede a describir el análisis realizado para la ejecución de la presente práctica, también, se mostrará el diseño desarrollado y la simulación del mismo.

#### Diseño:

En un comienzo se identificó que eran necesarios seis módulos aparte de la entidad principal para el diseño requerido. La conexión de los mismos se encuentra en la vista de la figura **0-1**.

El primero de estos módulos era el llamado counterpulse el cual se encargaba de indicar a la ALU el cambio de operación; es importante mencionar, que este módulo tiene instanciado otra entidad que recibe el nombre de pulse, el cual se encarga de generar un solo pulso por cada vez que se presione el botón de cambio de operación, ejerciendo un control en el cambio de la misma. El módulo **counterpulse** es secuencial debido a que solo realiza el cambio de operación con los flancos de subida de reloj. El segundo módulo es la ALU (ver imagen 0-2), la cual realiza todas las operaciones solicitadas (suma, resta, AND y OR), y por medio de un MUX determina cuál es la operación que debe ir a la salida. Además, por medio de un circuito definido por compuertas, puede determinar la salida ALUFlags. Vale aclarar que, este módulo tiene un Fulladder integrado, con el cual se realizan las operaciones de suma y resta, en caso de resta se ingresa el valor en complemento a2 y en caso de suma el valor se entrega tal cual ingresa.

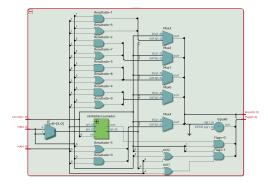


Figura **0-2**: ALU

El tercer módulo elaborado recibe el resultado de la ALU, esto con la intención de separar el valor en unidades, decenas, e igualmente se identifica si el resultado de la operación es un número con signo o sin signo para su posterior representación, lo anterior se realiza con el fin de que la señal pueda ser ingresada en los diferentes displays de 7 segmentos.

Finalmente, se tiene un módulo encargado de codificar los valores ingresados al display, en este módulo, también se evalúan ciertas condiciones específicas para poder mover el signo en caso de que el valor generado no tenga decenas, o para apagarlo en caso de no necesitar el display. Se usaron tres de los displays de 7 segmentos para observar el resultado de la ALU, asimismo, se utilizaron 8 de los LEDs de la tarjeta para mostrar las banderas y la operación a realizar.

#### Simulación

En el mismo archivo en el cual se diseñó la entidad principal, se creó el test bench con el propósito de poder evaluar el comportamiento del hardware antes de que este fuese implementado en la tarjeta. Esta simulación puede apreciarse en la siguiente imagen.

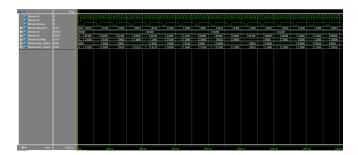


Figura **0-3**: Simulación

En la simulación mostrada, es posible observar las salidas de las operaciones hechas por la ALU después de pasar por el convertidor de decenas y unidades, así como el clk, que emula al reloj de la tarjeta, también, las banderas correspondientes a cada operación, el tipo de operación y las entradas ingresadas.

## **Conclusiones**

A pesar de que el objetivo principal era el desarrollo de la ALU, lo más complicado estuvo en los otros módulo que componían el proyecto; por un lado, el diseño del testbench resultó muy extenso en un comienzo debido a la gran cantidad de variables a alterar; sin embargo, el uso de un ciclo for, redujo en gran medida el código, simplificando el proceso, con esta herramienta fue posible que, mediante iteraciones, se pudiese variar todos los valores sin necesidad de definirlos manualmente. Por otro lado, se invirtió mucho tiempo en la solución de diferentes aspectos estéticos, ya que, con el fin de hacer que el resultado se visualizase de una manera agradable, se hicieron reiterados intentos para acomodar los valores y signos necesarios para el display de 7 segmentos; es importante resaltar, que la parte funcional estuvo lista con relativa prontitud.

Una vez más, se destaca la gran utilidad que ofrece la herramienta de simulación ModelSim, puesto que esta permitió realizar avances significativos y evaluar el comportamiento del diseño planteado.